one end (or any other form of identification for that matter). However, if desired, an arrow may be drawn pointing toward the object within which the other object is defined or to which the other object is assigned.

[0061] A descriptive name should appear within the symbol representing each object shown in an OED, but the detailed properties of the object preferably is not set forth in the OED. Any actual, detailed description of an object that the architect feels is necessary may be disclosed in another document, such as by a simple text description. This should help keep the OEDs from becoming too complicated. However, text descriptions of the details of objects can be placed within the symbols if deemed desirable. Alternately, a reference to the textual description can be placed within the symbol, e.g., "See Document 3.41". In yet another embodiment of the invention, in which the invention is implemented in a GUI by software, one may double click with one's pointer within the symbol in order to call up the textual description. The text description may appear in a separate pop-up window, for instance, like hyperlinking. Alternately, the text description may be called up via a mouse-over type retrieval. In an even further possible embodiment, the text description may be hidden text, which can be made visible in the same manner as the hidden comments feature provided in certain word processing programs such as IBM WordPro.

[0062] Next, all other objects defined within or assigned to the main object **501** preferably are drawn outside the large circle **503**. These include a button symbol for each of the buttons, including a "NEW" button symbol **502**, an "OPEN" button symbol **504**, a "FLIGHTS" button symbol **506**, a "CITIES" button symbol **508**, a "PHONES" button symbol **510**, a "PROGRAMS" button symbol **512**, a "NEWS" button symbol **514**, a "CALENDAR" button symbol **516**, and a "HELP" button symbol **518**. Since each button will cause something to happen, i.e., some piece of code to be executed, each button has associated with it an event script **520**, **522**, **524**, **526**, **528**, **530**, **532**, **534**, and **536**, respectively, to represent the event and the script that will cause that button to invoke code. In this particular example, for each of the buttons, the event is a single mouse click over the corresponding button. This is represented in the script object's label by the word "click." Furthermore, the method invoked thereby is identified by name in the label (e.g., "new file" for the new button). A description of the method invoked is not provided in the OED. As previously mentioned, it might be defined in a separate document or provided or referenced within the event script symbol by means of hidden text, hyperlinking, or otherwise.

[0063] In addition to the buttons discussed above, the FLIGHTS window **201***a* has a menu bar **221**. A mouse click on each menu will invoke some other program module. Menu symbol **521** represents the menu bar **221** in the FLIGHTS window. If the architect thinks that the menus are sufficiently complex and/or non-standard, he or she may provide a separate, additional OED for the menu bar. Alternately, the application architect simply may provide a textual description of the menus that comprise the menu bar (which could be provided in a separate document or embedded within the OED by any of the previously described techniques). As an even further alternative, the architect may have, instead, decided to show each menu in the menu bar as a separate menu symbol directly within the OED **300***a*, rather than as a single menu symbol representing the entire menu bar as shown. Any of these options (as well as many other options) for illustrating the desired logic would be sufficient to describe the logic to a programmer.

[0064] There is more than one way to represent the business logic in accordance with the present invention. In fact, for most, if not all GUI's, there will probably be any number of ways that an architect or documentor could use any single embodiment of the present invention to represent the logic.

[0065] A menu object in an OED does not mean that the menu will open the object to which it is connected (window **201***a* in this case), but that, when the window (or other object) is open, the menu will be available within that object.

[0066] In addition to the objects, i.e., buttons and menus, that can be seen on the web page, there are a plurality of methods that are available in the window. They are represented by BUILD FLIGHTS method symbol **553**, HANDLE PRINTING method symbol **555**, GET AIRPORT BY CODE method symbol **557**, VALIDATE CITY NAME method symbol **559**, VALIDATE DATE AND TIME method symbol **561**, VALIDATE AIRLINE method symbol **563**, and GET DATABASE CONNECTION method symbol **565**. As previously noted, linking a method object to a window object in an OED does not mean that the window will call that method. Rather, it means that the method is available in that window (as well as in any other window or other object that inherits the properties of that object). Thus, any event script assigned to that window or any other object within that window object can invoke that method.

[0067] Again, the functionality of the actual methods represented by the method symbols should be described in a separate document or by text hidden or otherwise embedded within the OED. However, by way of exemplification, for instance, the VALIDATE AIRLINE method might be a code module that checks the name of an airline entered by a user of the web page **200** to assure that the name corresponds to a known airline and, if it does not, inform the user of that fact and ask the user to enter a proper airline. As another example, the GET DATABASE method may connect to a database. A database connector is needed to build a list of potential flight plans for a user. The HANDLE PRINTING method may include, for instance, functions such as formatting the data, sending the data to a printer, monitoring the printer status, etc. The HANDLE PRINTING method object **389** is likely to be a complex object in the sense that it likely has many other objects assigned to or defined within it. Accordingly, an application architect may decide to prepare another OED to set forth the details of the HANDLE PRINTING object. If so, a reference to the other OED might be placed within the HANDLE PRINTING method symbol **555** in any of the manners previously mentioned.

[0068] Note that there is no button or menu item in the FLIGHTS window **201***a* that would likely be designed to invoke some of the methods shown in the diagram. For instance, the "validate airline" method most likely would be invoked in window **201***b*, for example, in response to a user tabbing out of one of the boxes shown to the right of the text AIRLINES after he or she enters an airline name in the box. Thus, the architect might reasonably have decided not to place the VALIDATE AIRLINE function symbol in the OED